

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 1月23日

出 願 番 号

Application Number:

特願2003-015301

[ST.10/C]:

[JP2003-015301]

出 願 人

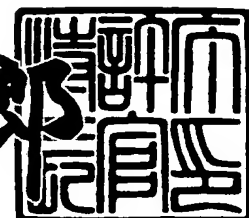
Applicant(s):

インターナショナル・ビジネス・マシーンズ・コーポレーション

2003年 5月20日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3037024

【書類名】 特許願

【整理番号】 JP9020242

【提出日】 平成15年 1月23日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 7/38  
G06F 7/44  
G09C 1/00

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 佐藤 証

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 高野 光司

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

【復代理人】

【識別番号】 100104880

【弁理士】

【氏名又は名称】 古部 次郎

【選任した復代理人】

【識別番号】 100118201

【弁理士】

【氏名又は名称】 千田 武

【手数料の表示】

【予納台帳番号】 081504

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0207860

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 乗算器及び暗号回路

【特許請求の範囲】

【請求項 1】 乗算対象である 2 つの入力値に対して部分積を求め、当該部分積を冗長 2 進形式で加え合わせるワラスツリー部と、

前記ワラスツリー部から出力される冗長 2 進数を 2 の補数形式に変換する桁上げ加算器とを備え、

前記ワラスツリー部は、

前記部分積の値を桁ごとに加算する合計値計算部と、

前記合計値計算部による加算における桁上がり値を加算する桁上がり値計算部と  
を備えることを特徴とする乗算器。

【請求項 2】 前記合計値計算部の計算結果を 2 の拡大体に対する乗算結果として出力することを特徴とする請求項 1 に記載の乗算器。

【請求項 3】 前記桁上げ加算器は、前記合計値計算部の計算結果と前記桁上がり値計算部の計算結果とを加算して整数に対する乗算結果として出力することを特徴とする請求項 1 に記載の乗算器。

【請求項 4】 前記合計値計算部は、前記部分積と共に他の所定の値を対応する桁ごとに加算することによって積和演算を行うことを特徴とする請求項 1 に記載の乗算器。

【請求項 5】 乗算対象である 2 つの入力値に対して部分積を求め、半加算器及び全加算器を用いて当該部分積を加算し、当該入力値の乗算を行う乗算器において、

前記部分積における合計値を各桁別に計算し、前記入力値が 2 の拡大体である場合に乗算結果として出力する乗算手段と、

前記乗算手段の計算において生じる桁上がり値を加算する桁上がり値加算手段と、

前記乗算手段の計算結果と前記桁上がり値加算手段の計算結果とを加算し、前記入力値が整数である場合に乗算結果として出力する加算手段と

を備えることを特徴とする乗算器。

【請求項 6】 前記乗算手段は、前記半加算器及び前記全加算器から出力される XOR（排他的論理和）演算による加算項のみを集めて外部へ出力することを特徴とする請求項 5 に記載の乗算器。

【請求項 7】 前記桁上がり値加算手段は、前記乗算手段にて加算される加算項以外の全ての項を集めて半加算器及び全加算器により桁上がり項と加算項とを含めた加算を行うことを特徴とする請求項 6 に記載の乗算器。

【請求項 8】 前記乗算手段における部分積の加算項に他の加算項を加えることによって積和演算を行うことを特徴とする請求項 6 に記載の乗算器。

【請求項 9】 データの暗号化または復号化のための演算を行う演算手段と

前記演算手段による演算を制御する制御手段とを備え、

前記演算手段は、

半加算器及び全加算器を用いた乗算器であって、

演算対象である 2 つの入力値に対して部分積を求め、当該部分積を冗長 2 進形式で加え合わせるワラスツリー部と、

前記ワラスツリー部から出力される冗長 2 進数を 2 の補数形式に変換する桁上げ加算器とを備え、

前記ワラスツリー部は、

前記部分積の値を桁ごとに加算する合計値計算部と、

前記合計値計算部による加算における桁上がり値を加算する桁上がり値計算部と

を備えることを特徴とする暗号回路。

【請求項 10】 前記演算手段は、有限体  $GF(2^n)$  上の演算を行う場合に前記合計値計算部の計算結果を出力し、有限体  $GF(p)$  上の演算を行う場合に前記桁上げ加算器の計算結果を出力することを特徴とする請求項 9 に記載の暗号回路。

【請求項 11】 前記合計値計算部は、前記半加算器及び前記全加算器から出力される XOR（排他的論理和）演算による加算項のみを集めて前記演算手段

の外部へ出力することを特徴とする請求項 9 に記載の暗号回路。

【請求項 1 2】 前記桁上がり値計算部は、前記乗算手段にて加算される加算項以外の全ての項を集めて半加算器及び全加算器により桁上がり項と加算項とを含めた加算を行うことを特徴とする請求項 9 に記載の暗号回路。

【請求項 1 3】 データの暗号化または復号化のための演算を行う演算手段と、

前記演算手段による演算を制御する制御手段とを備え、

前記演算手段は、

乗算対象である 2 つの入力値に対して部分積を求め、半加算器及び全加算器を用いて当該部分積を加算し、当該入力値の乗算を行う乗算器であって、

前記部分積における合計値を各桁別に計算し、前記入力値が有限体  $GF(2^n)$  である場合に乗算結果として出力する乗算手段と、

前記乗算手段の計算において生じる桁上がり値を加算する桁上がり値加算手段と、

前記乗算手段の計算結果と前記桁上がり値加算手段の計算結果とを加算し、前記入力値が整数である場合に乗算結果として出力する加算手段とを備えることを特徴とする暗号回路。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、コンピュータのプロセッサなどに用いられる演算装置に関し、特に有限体演算を行う乗算装置に関する。

【0 0 0 2】

【従来の技術】

今日、公開鍵暗号の方式としては、べき乗剰余演算（有限体  $GF(p)$  上の演算、なお  $GF(p)$  は整数を素数  $p$  で除算した余りの集合）を利用した RSA (Rivest, Shamir, Adleman) 方式が主流となっている。公開鍵暗号では鍵長（ビット数）が長いほど安全性が向上するが、この RSA 方式の暗号では実用上十分な安全性を得るために必要な鍵長が 5 1 2 ビットから、1 0 2 4 ビット、2 0 4 8 ビッ

トと増加してきており、演算時間と回路実装におけるリソースの増大が問題となっている。これに対し、楕円暗号を用いた公開鍵暗号方式（以下、楕円暗号方式）では、160ビットや224ビットの演算で、1024ビットや2048ビットのRSA方式と同等の安全性を確保することが可能である。

#### 【0003】

楕円暗号の基本演算は、RSA暗号と同じ $GF(p)$ をベースにした演算と、2の拡大体 $GF(2^n)$ による演算の2つに大別される。 $GF(2^n)$ の演算は、XOR (exclusive OR排他的論理和) が基本であり、加算時に桁上がりを生じないため、整数の剰余演算による $GF(p)$ に対して高速である。しかしながら、 $GF(2^n)$ 上の楕円暗号方式においても、最も重要なアルゴリズムの1つである、楕円DSA（デジタル署名アルゴリズム）による署名をサポートするためには $GF(p)$ 上の乗剰余演算も必要とされる（例えば、非特許文献1参照）。

#### 【0004】

なお、乗剰余演算を高速に実行するアルゴリズムにモンゴメリ乗算がある（例えば、非特許文献2参照）。初期のアルゴリズムは $GF(p)$ に対するものであり、加算器ベースのアルゴリズムであったが、今日では乗算器を使用したアルゴリズムや $GF(2^n)$ 上の演算への拡張がなされている。

#### 【0005】

##### 【非特許文献1】

今井秀樹著、「符号理論」、電子情報通信学会、1990年

##### 【非特許文献2】

Johann Groszschädl, "A Bit-Serial Unified Multiplier Architecture for Finite Fields  $GF(p)$  and  $GF(2^m)$ ", C.K. Koc, D. Naccache, and C. Paar (Eds.): CHES 2001, LNCS 2162, p.202-219, 2001. Springer-Verlag Berlin Heidelberg 2001.

##### 【非特許文献3】

I.F.Blake, N.P.Smart, G.Seroussi著、鈴木治郎訳、「楕円曲線暗号」、ピアソンエデュケーション、2001年

##### 【非特許文献4】

江藤良純、金子敏信 監修、「誤り訂正符号とその応用」、オーム社、1996年

【0006】

【発明が解決しようとする課題】

上述したように、公開鍵暗号を楕円暗号にて実現する場合、 $GF(p)$ による演算と $GF(2^n)$ による演算とを行うことが必要となる。したがって、暗号回路においては、 $GF(p)$ の演算を行う乗算器と $GF(2^n)$ の演算を行う乗算器とが必要となる。

ここで、8ビット程度の乗算器を用いる回路であれば、 $GF(p)$ の演算を行う乗算器と $GF(2^n)$ の演算を行う乗算器との両方を載せて、セレクタで切り替える構成としても回路規模に与える影響は小さい。しかし、高速化のために32ビットや64ビットの乗算器を使用する場合、乗算器のゲート数が増加するビット数の自乗のオーダーで増加し、またセレクタや配線も増大するため、 $GF(p)$ の演算を行う乗算器と $GF(2^n)$ の演算を行う乗算器とを両方搭載するとすれば、回路規模の増大が無視できない。

したがって、公開鍵暗号で扱われる160ビットや1024ビットといった数の演算を行う暗号回路の場合、暗号回路を搭載する機器や暗号回路自体の小型化を図るには、 $GF(p)$ の演算を行う乗算器と $GF(2^n)$ の演算を行う乗算器とを個別に搭載することは好ましくない。

【0007】

ところで、モンゴメリ乗算や楕円演算においては、 $GF(p)$ 上の演算と $GF(2^n)$ 上の演算とのアルゴリズムはほとんど同じである。そのため、回路化した場合のデータパスも、乗算コア自体を除けばほとんどそのまま共有することが可能である。上記の非特許文献2には、 $GF(p)$ 上の演算と $GF(2^n)$ 上の演算とに共用可能な乗算器が開示されている。

しかし、かかる従来の乗算器は、加算器をビット数分繰り返して使用し積を計算するシリアル乗算器であり、単に $GF(2^n)$ の演算時にキャリー (carry: 桁上がり) をゲートして無視 (Disable) するものである。シリアル乗算器で整数乗算を行うためには、毎サイクル毎に、キャリーを最下位ビット (LSB: Least



Significant Bit) から最上位ビット (MSB : Most Significant Bit) まで伝播させるか、冗長 2 進数のまま途中結果をビット数の 2 倍のレジスタに保存する必要がある。そのため、演算の高速化が困難であり、高速化するためには回路規模を大幅に増大しなければならない。

## 【 0 0 0 8 】

また上述したように、公開鍵暗号では 1 6 0 ビットや 1 0 2 4 ビットといった数が扱われ、その暗号回路では高速化のために非常に長い加算器が用いられる。そのため、データを高速に転送するためには、バス幅を加算器のビット長まで広げる必要があるため、物理的なチップサイズの増大を招来する。また、入力が 1 0 2 4 ビット  $\times$  2、出力が 1 0 2 4 ビットのバスともなると、汎用の ASIC 用ライブラリによる自動論理合成および配置配線では実装することができず、人手によるカスタムレイアウトという繁雑な作業を要することもある。一方、バス幅を広げないとすれば、処理の実行時に演算対象のデータがそろうまで加算器を待たせる制御を行わなくてはならず、実行性能の低下を招く。

## 【 0 0 0 9 】

以上、公開鍵暗号における暗号回路に用いられる乗算回路について論じたが、暗号に限らず、 $GF(p)$  上の演算と  $GF(2^n)$  上の演算とが要求される、符号理論を応用した種々のアプリケーションに関しても同様のことが言える。符号理論を応用した他のアプリケーションの例としては、誤り訂正符号による誤り訂正回路がある（例えば、非特許文献 3、4 参照）。

## 【 0 0 1 0 】

そこで本発明は、回路規模を増大することなく、通常の整数乗算器および  $GF(2^n)$  上の乗算器として使用することが可能なパラレル乗算回路を提供することを目的とする。

また本発明は、 $GF(p)$  及び  $GF(2^n)$  の 2 つの演算を要する楕円暗号や RSA 方式を初めとする乗剰余系の公開鍵暗号を 1 つのアクセラレータコアで実現する暗号回路を提供することを他の目的とする。

## 【 0 0 1 1 】

【課題を解決するための手段】

上記の目的を達成する本発明は、次のように構成された乗算器として実現される。すなわち、この乗算器は、乗算対象である2つの入力値に対して部分積を求め、この当該部分積を冗長2進形式で加え合わせるワラスツリー部と、このワラスツリー部から出力される冗長2進数を2の補数形式に変換する桁上げ加算器とを備え、ワラスツリー部は、部分積の値を桁ごとに加算する合計値計算部と、この合計値計算部による加算における桁上がり値を加算する桁上がり値計算部とを備えることを特徴とする。

かかる乗算器において、2の拡大体（有限体 $GF(2^n)$ ）に対する乗算を行う場合には、合計値計算部の計算結果のうち桁上がりを除く各桁の加算結果を当該乗算の結果として出力することができる。また、桁上げ加算器は、合計値計算部の計算結果と桁上がり値計算部の計算結果とを加算し、整数に対する乗算を行う場合には、この桁上げ加算器の計算結果を当該乗算の結果として出力することができる。

#### 【0012】

また、本発明は、乗算対象である2つの入力値に対して部分積を求め、半加算器及び全加算器を用いて部分積を加算し、入力値の乗算を行う、次のように構成された乗算器としても実現される。すなわち、この乗算器は、部分積における合計値を各桁別に計算し、入力値が2の拡大体である場合に乗算結果として出力する乗算手段と、この乗算手段の計算において生じる桁上がり値を加算する桁上がり値加算手段と、乗算手段の計算結果と桁上がり値加算手段の計算結果とを加算し、入力値が整数である場合に乗算結果として出力する加算手段とを備えることを特徴とする。

この乗算器において、乗算手段は、前記半加算器及び全加算器から出力されるXOR演算による加算項のみを集めて外部へ出力する。また桁上がり値加算手段は、乗算手段にて加算される加算項以外の全ての項を集めて半加算器及び全加算器により桁上がり項と加算項とを含めた加算を行う。

#### 【0013】

さらに本発明は、これらの乗算器を演算手段として搭載した種々の回路（アプリケーション）としても実現される。典型的な例としては、暗号回路における暗

号化または復号化処理のための演算を行う演算器として搭載することができる。

【0014】

【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいて、この発明を詳細に説明する。

本実施の形態は、パラレル乗算器において、 $GF(p)$ 上の演算と $GF(2^n)$ 上の演算とを実現する。 $GF(p)$ 上の乗算（整数乗算）を行うパラレル乗算器は、部分積（各桁の数ごとの積）をキャリーセーブ形式（冗長2進形式）で加え合わせていくワラスツリー（Wallace tree）部と、ワラスツリー出力の冗長2進数を2の補数形式に変換する桁上げ加算器とによって構成される。

【0015】

乗算器への $n$ ビットの2入力を $A = (a_{n-1}, \dots, a_1, a_0)$ 、 $B = (b_{n-1}, \dots, b_1, b_0)$ とすると、ワラスツリー部ではビットごとの部分積を半加算器（HA）と全加算器（FA）とによって加え合わせていく。

図1は共に8ビットの入力 $A$ 、 $B$ を例とした乗算のイメージを示す図、図2は図1の乗算において部分積を計算するための回路構成を示す図、図3は図2の回路で計算された部分積を加え合わせるための回路構成を示す図である。

図2に示す回路によって、 $n^2$ 個の部分積 $a_i b_j$ （ $i, j = 0, 1, 2, 3, 4, 5, 6, 7$ ）がそれぞれ算出され、図3に示す回路の対応する桁の全加算器または半加算器へ送られる。そして、図3に示す回路の出力（ $d_0, d_1, \sim, d_{15}$ ）が、桁上げ加算器を経て乗算結果として出力される。

【0016】

図4は半加算器の構成を示す図、図5は全加算器の構成を示す図である。

半加算器は、次の数1式のように2ビットの入力に対して $carry$ （桁上がり値）と $sum$ （合計値）の2ビットを計算する。

【数1】

$$\begin{cases} sum_{HA}(x_0, x_1) = x_0 \oplus x_1 \\ carry_{HA}(x_0, x_1) = x_0 \cdot x_1 \end{cases}$$

また、全加算器は、数2式のように3入力に対して $carry$ と $sum$ の2ビ

ットを計算する。

【数 2】

$$\begin{cases} \text{sum}_{\text{FA}}(x_0, x_1, x_2) = x_0 \oplus x_1 \oplus x_2 \\ \text{carry}_{\text{FA}}(x_0, x_1, x_2) = x_0 \cdot x_1 + x_1 \cdot x_2 + x_2 \cdot x_0 \end{cases}$$

上の各式において、「 $\cdot$ 」、「 $+$ 」は、それぞれAND（論理積）、OR（論理和）を意味している。また「 $\oplus$ 」に「 $+$ 」の演算記号はXOR（排他的論理和）を意味している。

【0017】

上記のような整数乗算回路に対して、 $GF(2^n)$ 上の乗算を行うパラレル乗算器は、 $\text{carry}$ は発生せず、ワラスツリー部において、部分積を桁ごとに全てXORするだけである。すなわち、図1と同様の8ビットの入力A、Bに対する演算を行う乗算回路において、ワラスツリー部は、部分積を計算する図2と同様の回路と、得られた部分積を桁ごとにXOR演算する回路とを備える。また、 $GF(2^n)$ 上の乗算では $\text{carry}$ が発生しないため、ワラスツリー部による部分積のXOR演算の結果が直ちに乗算結果であり、桁上げ加算器による変換を必要としない。

図6は、図2の回路で計算された部分積をXOR演算するための回路構成を示す図である。

【0018】

以上の点に鑑み、本実施の形態による乗算器は、上の数1、2式における $\text{sum}_{\text{HA}}()$ 及び $\text{sum}_{\text{FA}}()$ のXORを用いて、この部分に $GF(2^n)$ の乗算器を埋め込む構成とする。

図7は、本実施の形態による乗算器の構成を示す図である。

図7を参照すると、本実施の形態による乗算器100はワラスツリー部110と桁上げ加算器120とを備え、ワラスツリー部110は、入力値に対して桁ごとに合計値（ $\text{sum}$ ）と桁上がり値（ $\text{carry}$ ）とを計算する $\text{sum}$ 計算部111と、 $\text{sum}$ 計算部111の計算による桁上がり値を加算する $\text{carry}$ 計算部112と、 $\text{sum}$ 計算部111及び $\text{carry}$ 計算部112の計算結果を加算

して桁上げ加算器 1 2 0 に渡す H A / F A アレイ 1 1 3 とを備える。s u m 計算部 1 1 1 は、部分積の加算において半加算器及び全加算器から出力される X O R 演算による加算項だけを集めて出力するものである。また、c a r r y 計算部 1 1 2 は、s u m 計算部 1 1 1 にて加算される加算項以外の全ての項を集めて半加算器及び全加算器により桁上がり項と加算項とを含めた加算を行うものである。

桁上げ加算器 1 2 0 は、H A / F A アレイ 1 1 3 の出力を補数形式に変換して出力するもので、通常の平行乗算器における桁上げ加算器と同様である。

#### 【 0 0 1 9 】

すなわち、本実施の形態の乗算器 1 0 0 は、まず通常の整数乗算器と同様に、半加算器と全加算器とのアレイ (H A / F A array) からなる s u m 計算部 1 1 1 により c a r r y と s u m とを生成する。そして、通常の整数乗算器のワラスツリ一部では、同じ桁にある c a r r y と s u m とは区別せずに順次加算するのに対し、本実施の形態の乗算器 1 0 0 におけるワラスツリ一部 1 1 0 では、c a r r y 計算部 1 1 2 と s u m 計算部 1 1 1 とを用いて、c a r r y と s u m とを別のツリーで加算していく。

s u m 計算部 1 1 1 にて s u m だけを加算 (つまり X O R) していくことにより、 $2n-1$  ビットの  $G F(2^n)$  の乗算結果が得られる。最も深いパスでは、 $n$  ビットの部分積を 1 ビットに X O R 加算される。全加算器は 3 ビット入力に対して 1 ビットの s u m を出力するので、s u m 計算部 1 1 1 全体の遅延は、F A 換算で

#### 【 数 3 】

$$\lceil \log_3 n \rceil$$

段となる。

#### 【 0 0 2 0 】

一方、c a r r y 計算部 1 1 2 の回路構成は、様々な組み方が可能であるが、3 ビット入力に対して全加算器は 2 ビットの c a r r y と s u m とを出力するので、最大遅延は F A 換算で凡そ

【数 4】

$$\lceil \log_{3/2} n \rceil$$

段となる。

【0021】

図7を参照して明らかなように、本実施の形態による乗算器100は、GF(p)の乗算器の内部にGF(2<sup>n</sup>)の乗算器に相当するsum計算部111が埋め込まれた構成となっている。したがって、GF(2<sup>n</sup>)の乗算を行う場合は、図7に示すように、sum計算部111の計算結果をそのまま出力すれば良い。

上述したsum計算部111の遅延とcarry計算部112の遅延とから、sum計算部111の計算に要する時間の方が圧倒的に短い。そのため、図7に示すようにsum計算部111（GF(2<sup>n</sup>)の乗算器に相当する部分）を分離したことによりcarry計算部112が処理を待たされることはない。また、sum計算部111は、もともと整数乗算器に含まれていたXOR項を集めたものであるため、乗算器100全体を構成する回路が増加することもない。

すなわち、本実施の形態による乗算器100は、処理速度を低下させることなく、かつ回路規模を増大することなく、GF(p)の乗算およびGF(2<sup>n</sup>)の乗算が可能となっている。

【0022】

図8は、図1に示した8ビット×8ビットの乗算を行う本実施の形態の乗算器100におけるワラスツリー部110の構成例を示す図である。なお、図8では、半加算器を「H」と記述したブロック、全加算器を「F」と記述したブロックで表現しているが、これは図4、5にしたがっている。

図8に示すワラスツリー部110は、5つのステージからなる。また図8において、三角印(△)の入力は部分積、白丸印(○)の入出力はGF(p)及びGF(2<sup>n</sup>)の演算における合計値(sum、以下、第1sumと称す)、黒丸印(●)の入出力はGF(p)の演算のみにおける合計値(sum、以下、第2sumと称す)、四角印(■)の入出力はGF(p)の演算における桁上がり値(carry)を意味する。なお、図8に示すワラスツリー部110の構成は、説明の簡単

のため、多少の無駄を含んで表現されている。

【0023】

したがって、第1ステージでは各桁の部分積が入力され、`carry`と第1 `sum`とが出力される。

また、第2ステージでは第1ステージで得られた `carry` が加算されて第2 `sum` が出力されると共に、第1ステージで得られた第1 `sum` が加算されて `carry` と第1 `sum` とが出力される。ここで、第1ステージと第2ステージにおける第1 `sum` を加算する段とが、図7に示したワラスツリー部110の `sum` 計算部111に対応している。すなわち、この第2ステージで出力される第1 `sum` が、 $GF(2^n)$  の乗算結果である。

【0024】

また、第3ステージでは第2ステージで得られた `carry` 及び第2 `sum` が加算されて、加算結果の `carry` 及び第2 `sum` が出力される。そして、第2ステージで得られた第1 `sum` が通過している。この第3ステージにおける `carry` 及び第2 `sum` を加算する段が、図7に示したワラスツリー部110の `carry` 計算部112に対応している。

第4ステージでは、第3ステージで得られた `carry` 及び第2 `sum` と、第2ステージで得られ第3ステージを通過した第1 `sum` とが加算され、加算結果の `carry` 及び第2 `sum` が出力される。また、下位のいくつかの桁では第1 `sum` あるいは第2 `sum` が加算されずに通過している。この第4ステージが、図7に示したワラスツリー部110のHA/FAアレイ113に対応している。

第5ステージでは、第4ステージで得られた `carry` 及び第2 `sum` と、第4ステージを通過した第1 `sum` 及び第2 `sum` とが加算されて、加算結果である第2 `sum` が出力される。この第5ステージが、図7に示した桁上げ加算器120に対応している。すなわち、この第5ステージで出力される第2 `sum` が、 $GF(p)$  の乗算結果である。

【0025】

次の数5式は、任意の  $n$  ビット乗算において、本実施の形態の乗算器100における第1ステージの部分を表現したものである。

## 【数 5】

$$\begin{aligned}
& g_{k,j} = a_i \cdot b_j \quad (k=i+j, 0 \leq i, j < n) \\
& \{s_{k,0}, \dots, s_{k, \lceil (n-|k-n|)/3 \rceil}\} \\
& = \begin{cases} \{\text{sum}_{FA}(g_{k,0}, g_{k,1}, g_{k,2}), \dots, g_k\} & (|n-k| \bmod 3 = 0) \\ \{\text{sum}_{FA}(g_{k,0}, g_{k,1}, g_{k,2}), \dots, \text{sum}_{HA}(g_{k,k-1}, g_k)\} & (|n-k| \bmod 3 = 1) \\ \{\text{sum}_{FA}(g_{k,0}, g_{k,1}, g_{k,2}), \dots, \text{sum}_{FA}(g_{k,k-2}, g_{k,k-1}, g_k)\} & (|n-k| \bmod 3 = 2) \end{cases} \\
& \{c_{k,0}, \dots, c_{k, \lceil (n-|k-n|)/3 \rceil}\} \\
& = \begin{cases} \{\text{carry}_{FA}(g_{k,0}, g_{k,1}, g_{k,2}), \dots, \text{carry}_{HA}(g_{k,k-1}, g_k)\} & (|n-k| \bmod 3 = 1) \\ \{\text{carry}_{FA}(g_{k,0}, g_{k,1}, g_{k,2}), \dots, \text{carry}_{FA}(g_{k,k-2}, g_{k,k-1}, g_k)\} & (|n-k| \bmod 3 = 0 \text{ or } 2) \end{cases}
\end{aligned}$$

## 【0026】

なお、上記のように構成された本実施の形態の乗算器100は、演算対象である入力A ( $a_{n-1}, \dots, a_1, a_0$ )、B ( $b_{n-1}, \dots, b_1, b_0$ ) の部分積  $a_i b_j$  以外に別の加算項を1つあるいは複数組み込んで、ワラスツリー部110を構成することが容易である。したがって、例えば入力A、Bの乗算に入力Cを加算する積和演算機能も簡単に実現することができる。

図9は、図1に示した8ビットの入力A、Bの乗算に、同じく8ビットの入力Cを対応する桁ごとに加算する積和演算のイメージを示す図である。また図10は、図9に示した8ビットの積和演算を行う本実施の形態の乗算器におけるワラスツリー部110の構成例を示す図である。図10における半加算器および全加算器の表現は図8の場合と同様である。

図10のワラスツリー部110と図8のワラスツリー部110とを比較すると、 $A \times B$ における下位8桁の部分積に入力C ( $c_7, c_6, \dots, c_0$ )を加算するための加算器が追加されているものの、各ステージでの処理の内容は同様であることが分かる。

## 【0027】

以上説明した本実施の形態の乗算器100は、1つの乗算器100でGF(p)及びGF( $2^n$ )の2つの演算を行うことができるため、乗算器100を組み込む回路では、回路規模を増大することなく、これらの演算を実装することが可能である。



また本実施の形態の乗算器 1 0 0 は、 $n$  ビット  $\times$   $n$  ビットの平行乗算器であるため、例えば  $n = 32$  ビットバスとした場合、1 0 2 4 ビット加算器によるシリアル乗算器と同程度の処理を、より高速に行うことが可能である ( $32 \times 32 = 1024$ )。したがって、演算対象であるデータのサイズ (データビット長) の変化に対して柔軟性及び回路量当たりの処理能力が非常に高い。

## 【0028】

さらに本実施の形態の乗算器 1 0 0 は、 $GF(p)$  及び  $GF(2^n)$  の 2 つの演算を 1 つの乗算器 1 0 0 で行うことが可能であるため、符号理論を応用した種々のアプリケーションにて使用可能である。最も典型的な例としては、楕円暗号方式による公開鍵暗号を実現する暗号回路の演算器として用いられる。

図 1 1 は、本実施の形態による乗算器 1 0 0 を組み込んだ公開鍵暗号回路の構成例を示す図である。

図 1 1 に示す暗号回路は、データの暗号化または復号化の処理 (以下、暗号処理) における制御手段としてのデータ長カウンタ 2 1 0、暗号制御回路 2 2 0、鍵シフトレジスタ 2 2 1、アドレスレジスタ 2 3 0 及びメモリ制御回路 2 4 0 と、処理対象であるデータを保持する保持手段としてのメモリ 2 5 1、2 5 2 と、暗号処理の実行手段である積和演算器 2 6 0 とを備える。

## 【0029】

この暗号回路では、データ長カウンタ 2 1 0 にて指定されるデータサイズで、暗号制御回路 2 2 0 及び鍵シフトレジスタ 2 2 1 により、データの暗号化または復号化に用いられるパラメータ (鍵等) が設定される。そして、アドレスレジスタ 2 3 0 及びメモリ制御回路 2 4 0 の制御により、メモリ 2 5 1、2 5 2 に保持されている入力データ及び暗号制御回路 2 2 0 にて設定されたパラメータが積和演算器 2 6 0 に渡される。積和演算器 2 6 0 は、当該入力データの暗号化または復号化のための演算処理を行う。この積和演算器 2 6 0 として、本実施の形態の乗算器 1 0 0 を用いることができる。

## 【0030】

この暗号回路において、積和演算器 2 6 0 は、暗号処理において、 $GF(p)$  上の演算と  $GF(2^n)$  上の演算とを実行するものとする。例えば、楕円暗号方式に

において D S A 署名をサポートする場合や、楕円暗号方式と R S A 方式を併用する場合などである。

R S A 方式の暗号処理における演算や楕円暗号方式の D S A 署名では  $GF(p)$  上の演算を要するので、積和演算器 260 を構成する乗算器 100 の桁上げ加算器 120 の出力が演算結果として出力される。

一方、楕円暗号方式の暗号処理における演算では  $GF(2^n)$  上の演算を要するので、積和演算器 260 を構成する乗算器 100 の  $sum$  計算部 111 の出力が演算結果として出力される。

#### 【0031】

また、本実施の形態の乗算器 100 を使用可能な他のアプリケーションとして、誤り訂正回路がある。

図 12 は、本実施の形態による乗算器 100 を組み込んだ誤り訂正回路の構成例を示す図である。

図 12 に示す誤り訂正回路は、シンδροーム計算部 310 と、多項式生成部 320 と、誤り位置評価部 330 と、誤り値評価部 340 と、誤り訂正部 350 とを備える。ここで、シンδροーム計算部 310 は、データの誤りパターンに応じたシンδροームを生成する。また、データの消失があった場合はその位置を決定する。多項式生成部 320 は、シンδροームとデータ消失位置の情報から、誤り位置を求める多項式（誤り位置多項式）と誤りのパターンを求めるための多項式（誤り値多項式）とを生成する。誤り位置評価部 330 は、多項式生成部 320 にて生成された誤り位置多項式に基づいて誤り位置を計算する。誤り値評価部 340 は、誤り位置評価部 330 で求めた誤り位置と多項式生成部 320 にて生成された誤り値多項式とに基づいて誤り値を計算する。誤り訂正部 350 は、誤り位置評価部 330 で求めた誤り位置に対応するデータに、誤り値評価部 340 で求めた誤り値を排他的論理和することによって、誤りを訂正する。

#### 【0032】

この誤り訂正回路において、シンδροーム計算部 310、多項式生成部 320、誤り位置評価部 330 及び誤り値評価部 340 を構成する積和演算回路 311、321、322、331、341、342 及び除算回路 343 として、本実施

の形態の乗算器 1 0 0 を用いることができる。そして、 $GF(2^n)$ 上の演算結果が乗算器 1 0 0 の sum 計算部 1 1 1 から出力される。

【0 0 3 3】

【発明の効果】

以上説明したように、本発明によれば、回路規模を増大することなく、通常の整数乗算器および $GF(2^n)$ 上の乗算器として使用することが可能な乗算回路を提供することができる。

【0 0 3 4】

また、本発明による乗算回路は、符号理論を応用した種々のアプリケーションにて使用可能であり、この乗算回路を実装することにより、例えば、 $GF(p)$ 及び $GF(2^n)$ の2つの演算を要する楕円暗号やRSA方式を初めとする乗剰余系の公開鍵暗号を1つのアクセラレータコアで実現する暗号回路を提供することができる。

【図面の簡単な説明】

【図 1】 8ビットの入力を例とした乗算のイメージを示す図である。

【図 2】 図 1 の乗算において部分積を計算するための回路構成を示す図である。

【図 3】 図 2 の回路で計算された部分積を加え合わせるための回路構成を示す図である。

【図 4】 半加算器の構成を示す図である。

【図 5】 全加算器の構成を示す図である。

【図 6】 図 2 の回路で計算された部分積をXOR演算するための回路構成を示す図である。

【図 7】 本実施の形態による乗算器の構成を示す図である。

【図 8】 図 1 に示した8ビット×8ビットの乗算を行う本実施の形態の乗算器におけるワスツリー部の構成例を示す図である。

【図 9】 図 1 に示した8ビットの乗算に、他の8ビットの入力を加算する積和演算のイメージを示す図である。

【図 1 0】 図 9 に示した8ビットの積和演算を行う本実施の形態の乗算器

におけるワラスツリー部の構成例を示す図である。

【図 1 1】 本実施の形態による乗算器を組み込んだ暗号回路の構成例を示す図である。

【図 1 2】 本実施の形態による乗算器を組み込んだ誤り訂正回路の構成例を示す図である。

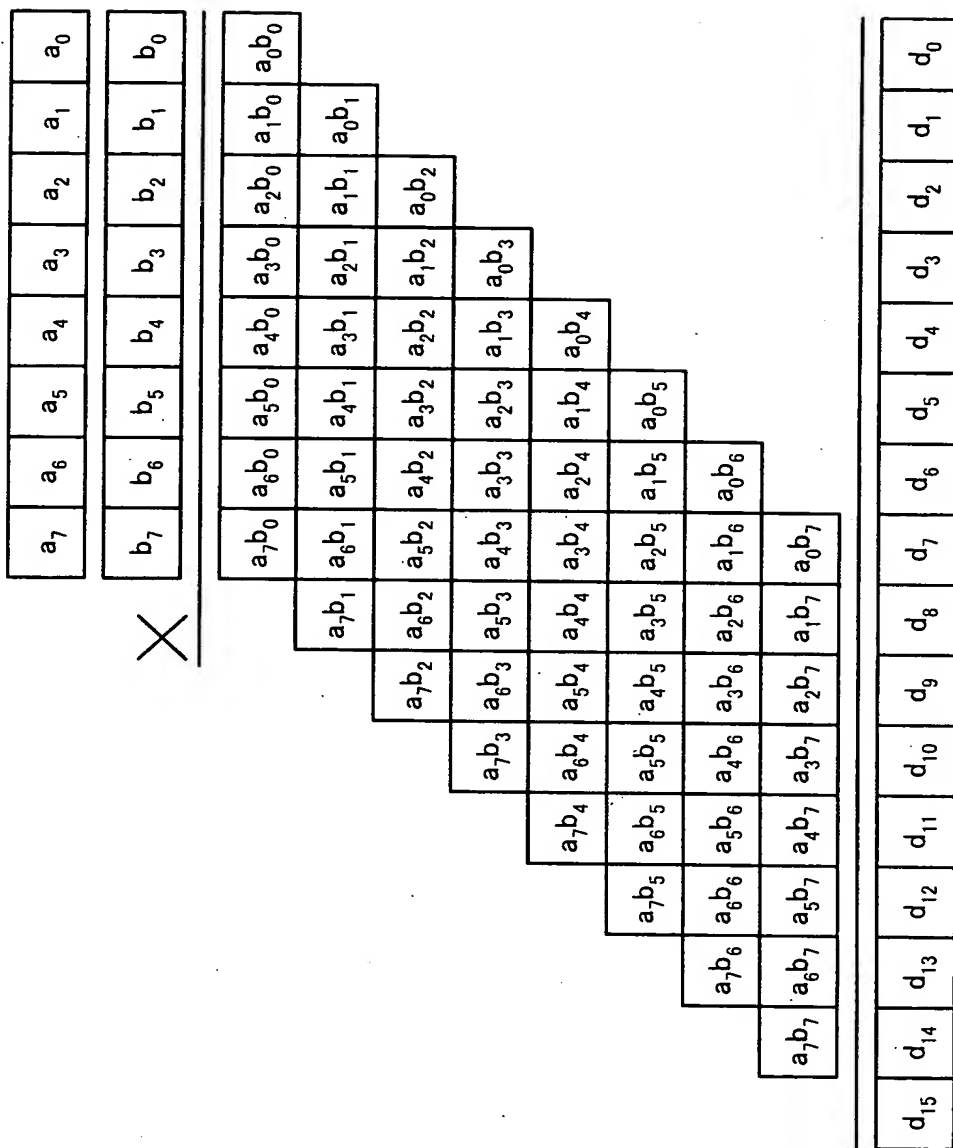
【符号の説明】

1 0 0 …乗算器、 1 1 0 …ワラスツリー部、 1 1 1 … s u m 計算部、 1 1 2 … c  
a r r y 計算部、 1 1 3 … H A / F A アレイ、 1 2 0 …桁上げ加算器

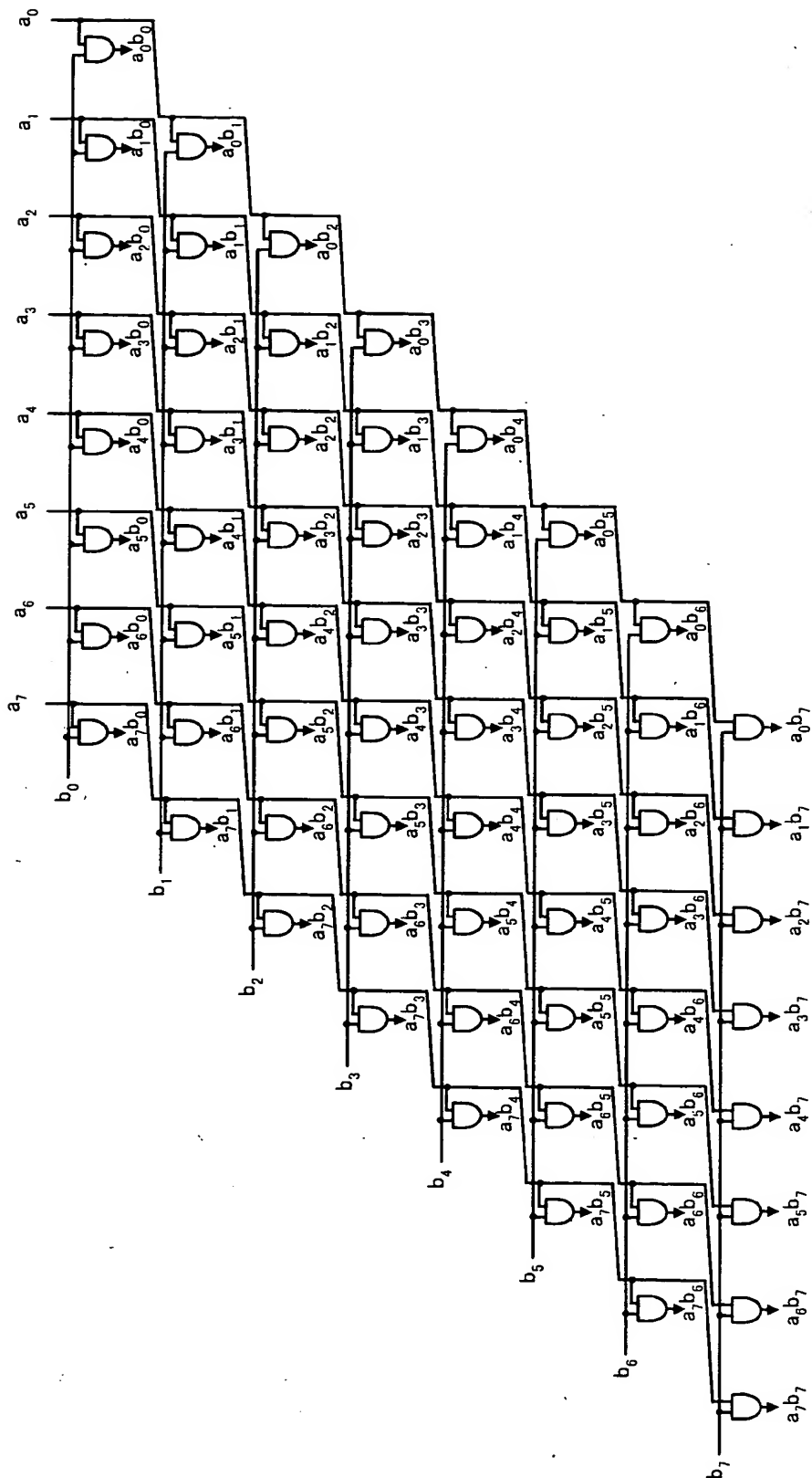
【書類名】

図面

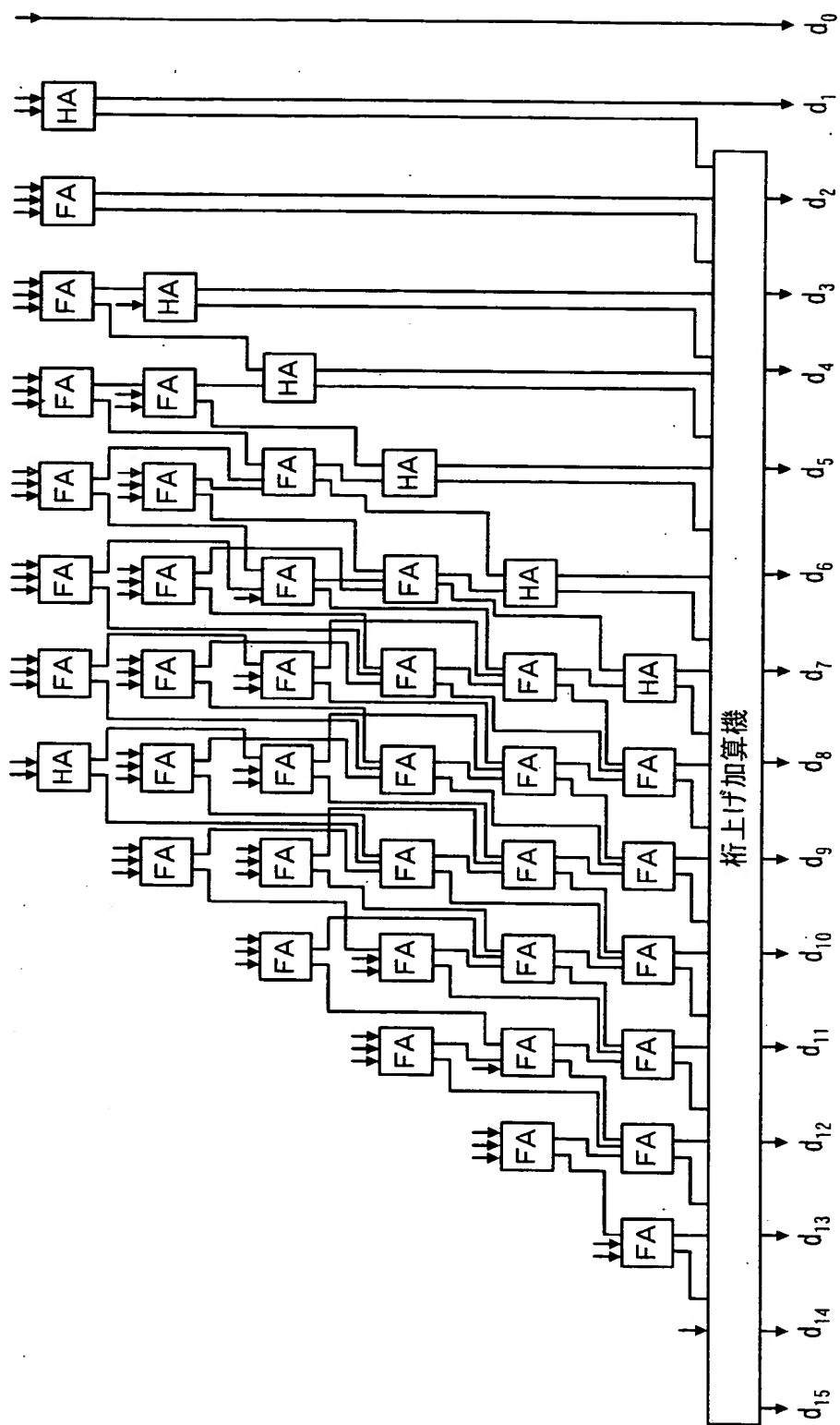
【図 1】



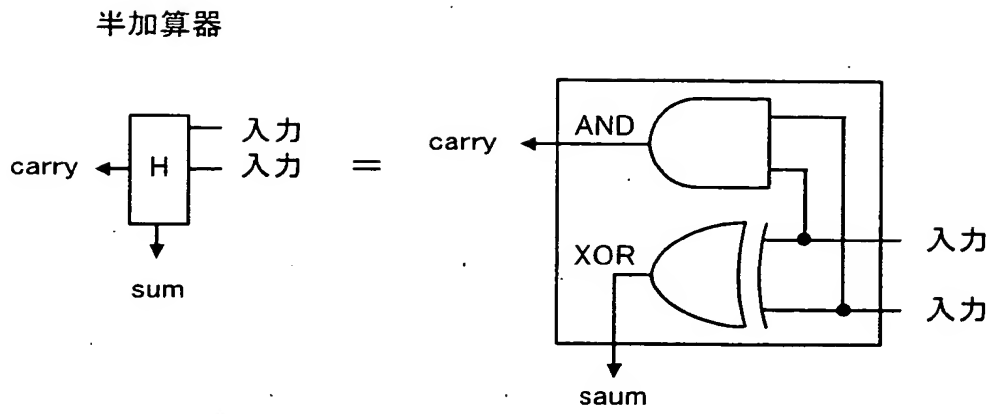
【図 2】



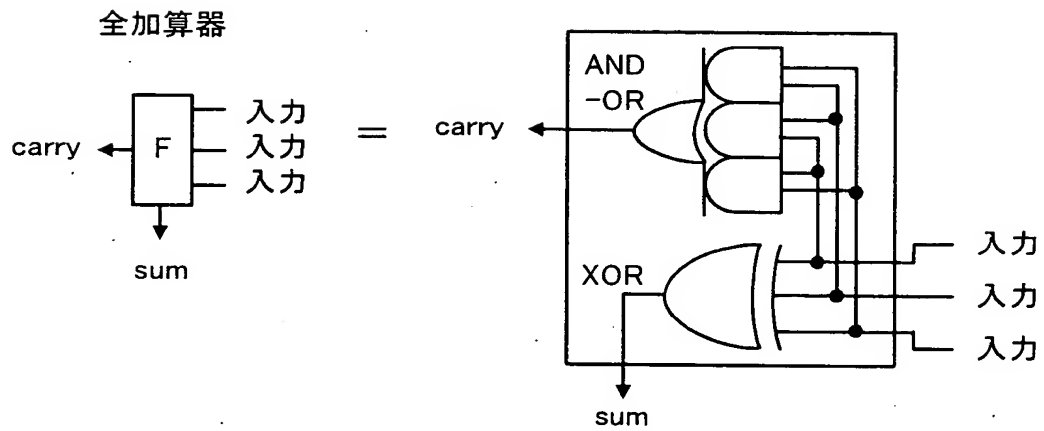
【図 3】



【図 4】

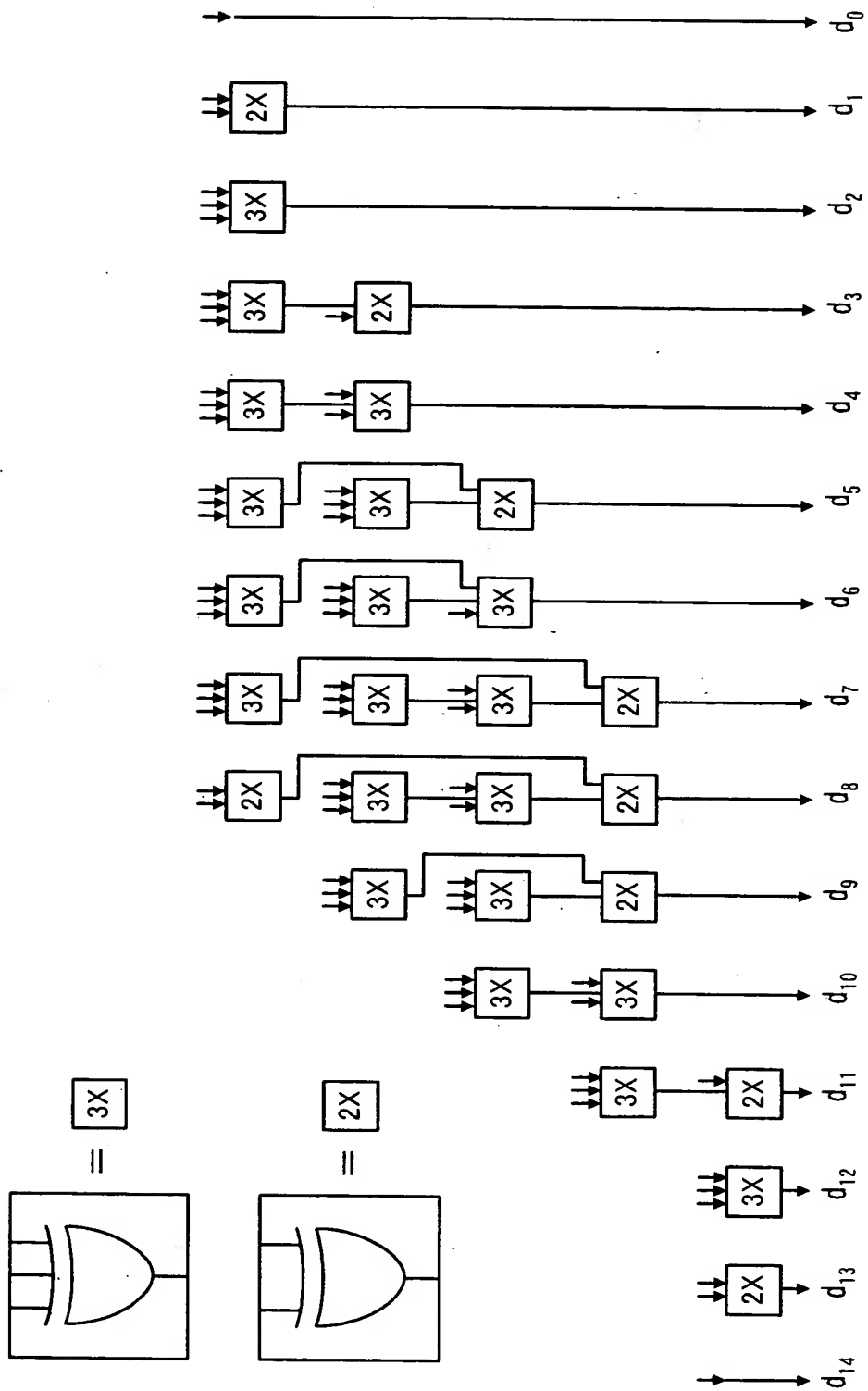


【図 5】

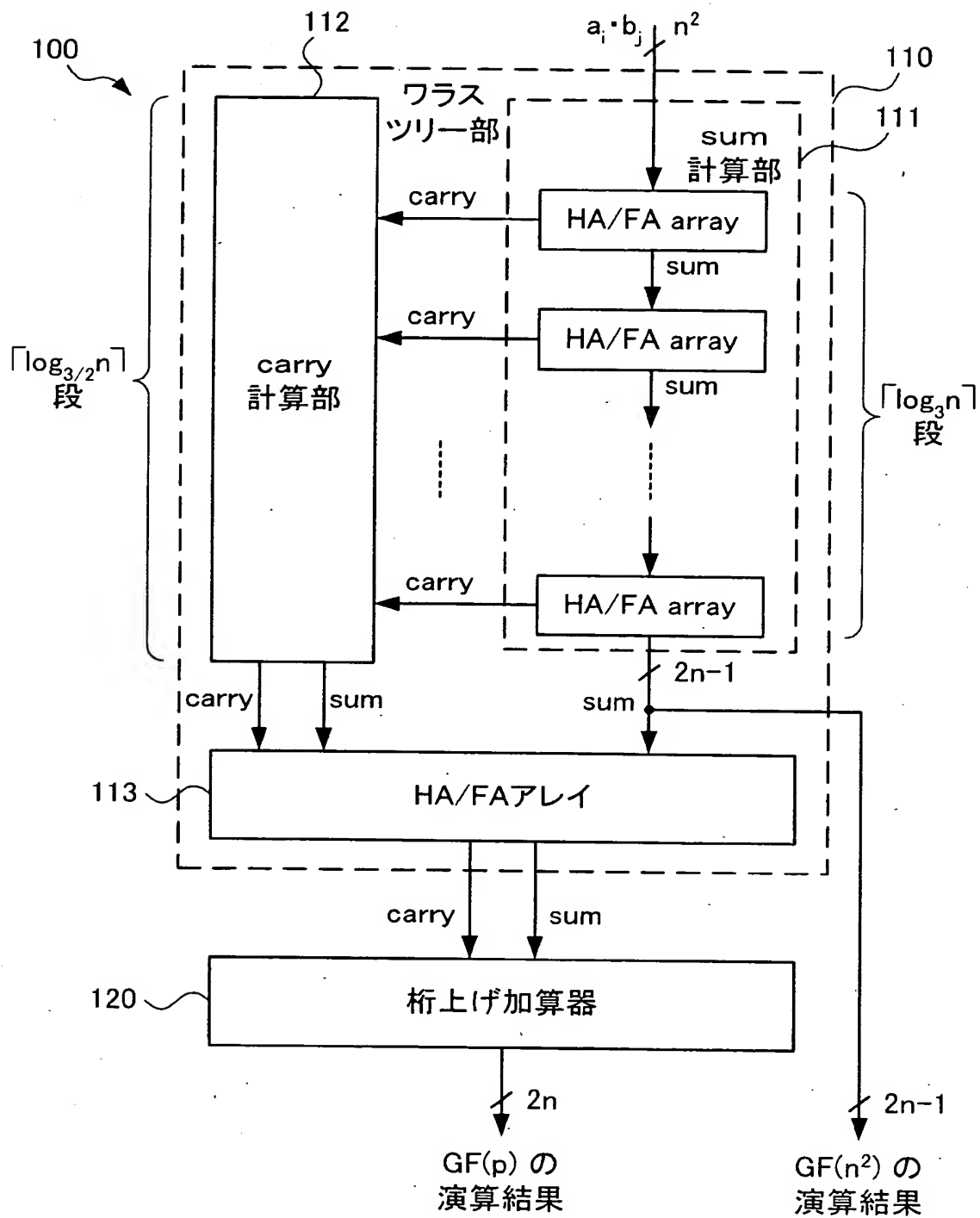




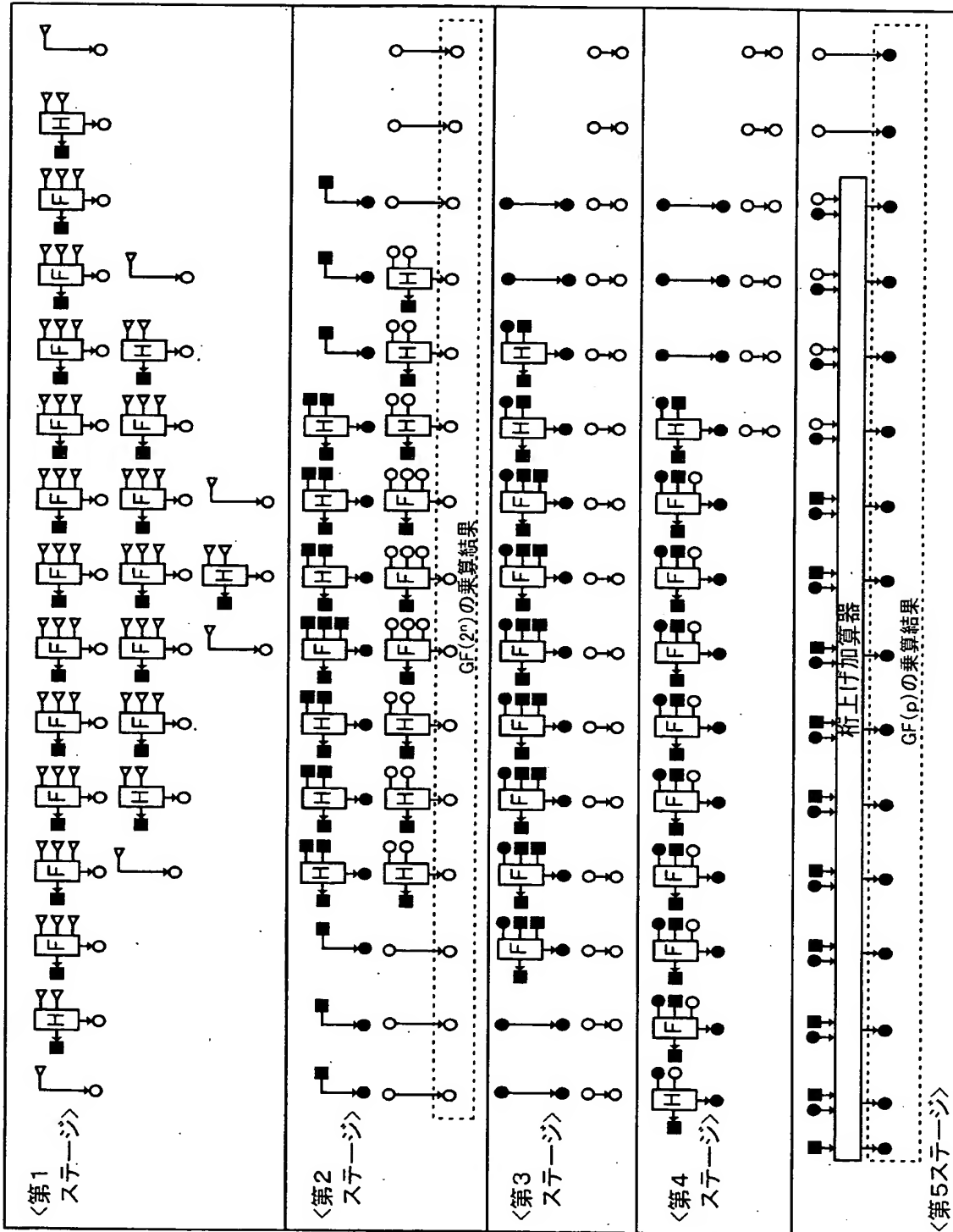
【図 6】



【図 7】



【図 8】



【図9】

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$
$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$

×

+

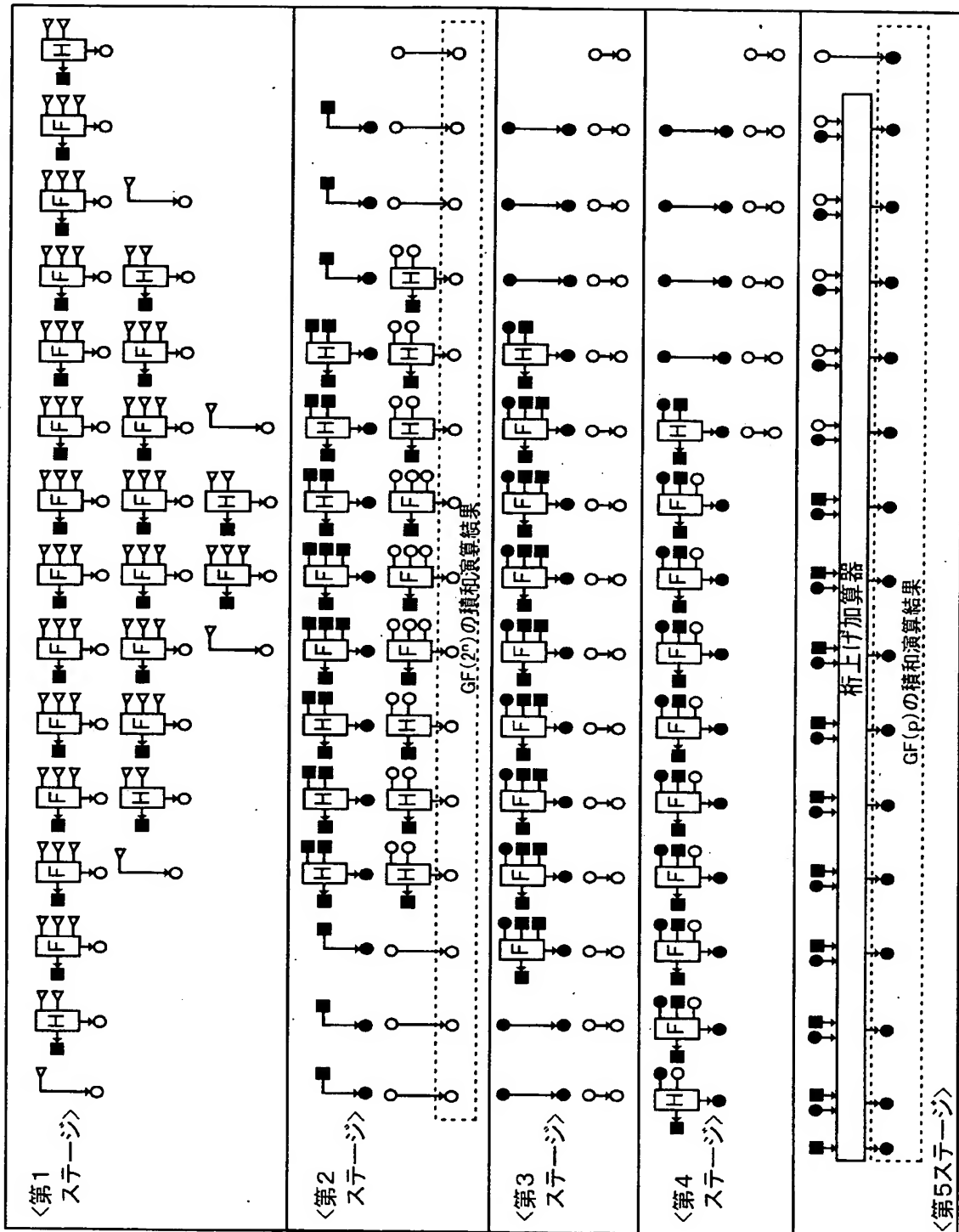
---

$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$
$a_7b_0$	$a_6b_0$	$a_5b_0$	$a_4b_0$	$a_3b_0$	$a_2b_0$	$a_1b_0$	$a_0b_0$
$a_7b_1$	$a_6b_1$	$a_5b_1$	$a_4b_1$	$a_3b_1$	$a_2b_1$	$a_1b_1$	$a_0b_1$
$a_7b_2$	$a_6b_2$	$a_5b_2$	$a_4b_2$	$a_3b_2$	$a_2b_2$	$a_1b_2$	$a_0b_2$
$a_7b_3$	$a_6b_3$	$a_5b_3$	$a_4b_3$	$a_3b_3$	$a_2b_3$	$a_1b_3$	$a_0b_3$
$a_7b_4$	$a_6b_4$	$a_5b_4$	$a_4b_4$	$a_3b_4$	$a_2b_4$	$a_1b_4$	$a_0b_4$
$a_7b_5$	$a_6b_5$	$a_5b_5$	$a_4b_5$	$a_3b_5$	$a_2b_5$	$a_1b_5$	$a_0b_5$
$a_7b_6$	$a_6b_6$	$a_5b_6$	$a_4b_6$	$a_3b_6$	$a_2b_6$	$a_1b_6$	$a_0b_6$
$a_7b_7$	$a_6b_7$	$a_5b_7$	$a_4b_7$	$a_3b_7$	$a_2b_7$	$a_1b_7$	$a_0b_7$

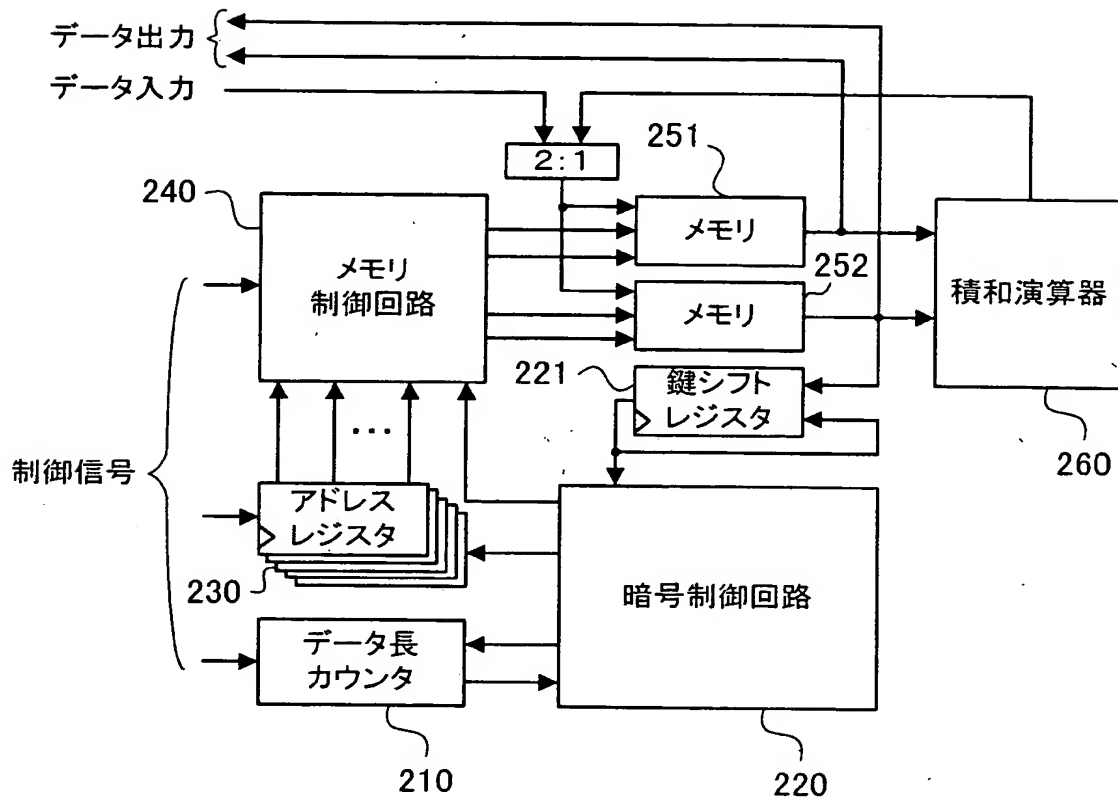
---

$d_{15}$	$d_{14}$	$d_{13}$	$d_{12}$	$d_{11}$	$d_{10}$	$d_9$	$d_8$	$d_7$	$d_6$	$d_5$	$d_4$	$d_3$	$d_2$	$d_1$	$d_0$
----------	----------	----------	----------	----------	----------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

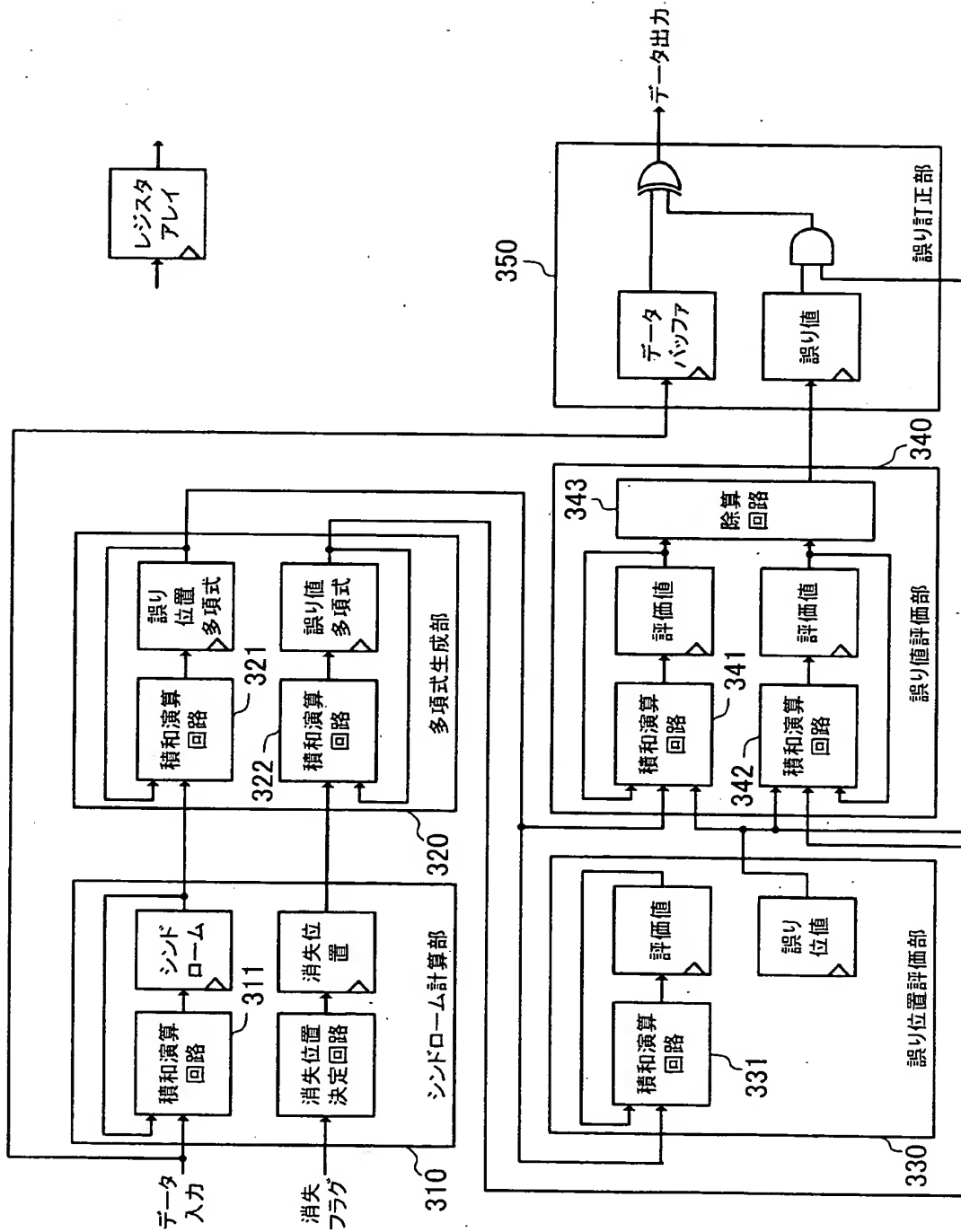
【図10】



【図 11】



【図 12】



【書類名】 要約書

【要約】

【課題】 回路規模を増大することなく、通常の整数乗算器および有限体  $GF(2^n)$  上の乗算器として使用することが可能な乗算回路を提供する。

【解決手段】 ワラスツリー部 110 と桁上げ加算器 120 とを備えた乗算回路であり、ワラスツリー部 110 は、部分積の値を桁ごとに加算する  $sum$  計算部 111 と、この  $sum$  計算部 111 による加算における桁上がり値を加算する  $carry$  計算部 112 とを備える。そして、2 の拡大体（有限体  $GF(2^n)$ ）に対する乗算を行う場合には、 $sum$  計算部 111 の計算結果を出力する。また、桁上げ加算器 120 は、 $sum$  計算部 111 の計算結果と  $carry$  計算部 112 の計算結果とを加算し、整数（有限体  $GF(p)$ ）に対する乗算を行う場合には、この桁上げ加算器 120 の計算結果を出力する。

【選択図】 図 7



認定・付加情報

特許出願の番号	特願 2003-015301
受付番号	50300107181
書類名	特許願
担当官	土井 恵子 4264
作成日	平成15年 1月24日

<認定情報・付加情報>

【特許出願人】

【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博

【代理人】

【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏

【代理人】

【識別番号】	100108501
【住所又は居所】	神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権
【氏名又は名称】	上野 剛史

【復代理人】

【識別番号】	100104880
【住所又は居所】	東京都港区赤坂5-4-11 山口建設第2ビル 6F セリオ国際特許事務所
【氏名又は名称】	古部 次郎

【選任した復代理人】

【識別番号】	100118201
--------	-----------

次頁有

認定・付加情報（続き）

【住所又は居所】 東京都港区赤坂 5-4-11 山口建設第二ビル  
6F セリオ国際特許事務所  
【氏名又は名称】 千田 武

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2002年 6月 3日

[変更理由] 住所変更

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニ  
ュー オーチャード ロード

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーショ  
ン